

Cryptography CS 555

Lecture 1



Overview of the Course

See the Course Homepage

- <http://www.cs.purdue.edu/homes/ninghui/courses/Fall04/index.html>

Goals of Cryptography

- The most basic problem: ensure security of communication over insecure medium
- Security goals:
 - privacy (secrecy, confidentiality)
 - only the intended recipient can see the communication
 - authenticity (integrity)
 - the communication is generated by the alleged sender

Approaches to Secure Communication

- Steganography
 - “covered writing”
 - hides the existence of a message
- Cryptography
 - “hidden writing”
 - hide the meaning of a message

Basic Terminology in Cryptography

- plaintexts, ciphertexts, keys
- encryption
- decryption
- cryptography
- cryptanalysis
- cryptology

A Sample List of Other Objectives

- Pseudo-random number generation
- Non-repudiation: Digital signatures
- Zero-knowledge proof
- Commitment schemes
- E-voting
- Secure Multi-party Computation (Secure Function Evaluation)

What Cryptography is About?

- Constructing and analyzing **protocols** which enables **parties** to achieve objectives, overcoming the influence of **adversaries**.
 - a protocol (or a scheme) is a suite of algorithms that tell each party what to do
- How to devise and analyze protocols
 - understand the **threats** posed by the adversaries and the **goals**

The Rules of the Game

1. Overcome the adversary only by means of protocols
2. Protocols are made public, only keys are secret
 - security by obscurity does not work

Phases in Cryptography's development

- First stage, paper and ink based scheme
- Second stage, use cryptographic engine
- Modern cryptography
 - relying on mathematics and computers
 - information-theoretic security (perfect secrecy)
 - computational security

Cryptoanalysis-driven design

- Cryptoanalysis-driven design
 1. A cryptographic goal is recognized
 2. Designs a solution
 3. Searches for an attack
 4. When an attack is found, go back to 2.
- Difficulty
 - One never knows if things are right
 - Cryptanalysis takes great cleverness; it is not clear that it can be effectively taught
 - More often an art, rather than a science

Computational Security

- Modern cryptography seeks security as long as adversaries don't have “too much” computing resources
- Takes cryptography from information theory to computer science
- A sample security statement:
 - assuming the adversary uses no more than t computing cycles, his probability of breaking the scheme is at most $t/2^{200}$

Atomic Primitives

- Atomic primitives are the most basic protocols used as building blocks for other protocols
 - block ciphers (DES algorithm)
 - one-way function: $f: D \rightarrow R$
 - f is easy to compute
 - f is hard to invert
 - examples (they are **believed** to be one-way function):
 - multiplication $f(a,b) = ab$
 - discrete log

The Provable-Security Approach

- Designing and analyzing primitives is based on the cryptanalysis-driven approach
 - important and challenging, but not the focus of this course
- Designing and analyzing protocols based on primitives
 - weak link in the “real-world” cryptography
 - focus of this course

Example Problems We Consider

- What is the best way to encrypt a large text file using DES, assuming that DES is “secure”
- What is the best way to design a signature scheme using multiplication, assuming that multiplication is one-way?
- How secure are known methods for these tasks?
- How to formalize such questions so that they can be answered?

What is This Course About?

- Mostly theory
 - understand the fundamentals of protocol design
 - understand the power and limitation of modern cryptography

Backgrounds Necessary for the Course

- Probability theory
 - a brief overview will be given to refresh your memory
- Algorithms and complexity
- Mathematical maturity
 - understand what is (and what is not) a proper definition
 - know how to write a proof
- Programming
 - for the project